# Computation of electrical field - Project 1

Ondřej Staněk

2012-12-14

**Project 1:** Given a plate capacitor we want the overall electric field including the edges with real behaviour: Compute E field inside plates nearby up to further away including a plot of bending at the edges. What can you tell about the potential?

**Abstract**

This paper describes models how to calculate electrical field for a plate capacitor. At first, author introduces numerical method for calculation the electric field of a plate capacitor. A discrete model of electric field calculation for any arbitrary charge distribution is described. Based on the observation of the discrete model, a generalization is made, which brings analytical expression for any arbitrary charge distribution. Finally, the derived analytic equations are adapted to describe field of a plate capacitor. The analytic expressions are then taken as a core of a Matlab program that visualizes the electrical field. For better understanding, the 2 dimensional case is considered first. Finally, the Matlab program is extended to support 3 dimensional visualization of the field.

# Part I
# Preliminaries

The whole work in this project is based on just only one simple equation for electrical field:

$$E = \frac{q}{4\pi\epsilon_0 r^2}$$

The equation states that the magnitude of electrical field $\vec{E}$ at distance $r$ from a point charge $q$ is proportional to the magnitude of charge $q$ divided by distance $r$ to the power of 2.

Moreover, please note that the vector $\vec{E}(\vec{v})$ lies on the line which connects point of interest $\vec{v}$ and location of the point charge $q$. Therefore, the electrical field $\vec{E}$ generated by a single point charge $q$ located at coordinates $\vec{p} = (p_x, p_y, p_z)$ can be described as follows:

$$\vec{E}(\vec{v}) = \frac{q}{4\pi\epsilon_0 \|\vec{v}-\vec{p}\|^2} \cdot \frac{(\vec{v}-\vec{p})}{\|\vec{v}-\vec{p}\|} \tag{1}$$

where the term $\frac{\vec{v}-\vec{p}}{\|\vec{v}-\vec{p}\|}$ is the normalized direction vector of the field and $\|\cdot\|$ is the euclidean norm on a vector space.

The second and last fact we need from the theory is the principle of superposition. Simply put, when there are more point charges present, the electrical fields generated by each of them are superposed (added) together to form one resulting electrical field.

So, this is all the theory we need for our models of electrical fields. Now we will just use basic rules from calculus and vector algebra to build various models for electrical field.

# Part II

# Models of electrical field

## 1 Discrete model of electrical field

My first attempt to solve the problem was to approximate electrical field of a capacitor by evaluating the electrical fields of many small point charges distributed evenly on the capacitor plates. Here I will describe a generalization how to calculate electrical field for any arbitrary distribution of small point charges. Assume we have a set of charges $s_i = (q_i, \vec{p_i})$, where $q_i$ is the charge in Columbs of a particular point and $\vec{p_i} = (p_{i,x} p_{i,y}, p_{i,z})$ is the coordinate vector describing the point charge position. Then, every point charge $s_i$ creates an electrical field component $\vec{E_i}$. Thanks to the principle of superposition, the resulting electrical vector field of all point charges is as follows:

$$\vec{E} = \sum_i \vec{E_i}$$

Denoting $\vec{E}(\vec{v})$ as the intensity of electrical field at point $\vec{v} = (v_x, v_y, v_z)$ and using equation (I) we get the formula for the total electrical field generated by all point charges:

$$\vec{E}(\vec{v}) \quad = \quad \sum_i \vec{E_i}(\vec{v}) = \sum_i \frac{q_i}{4\pi\epsilon_0 \|\vec{v}-\vec{p_i}\|^2} \cdot \frac{(\vec{v}-\vec{p_i})}{\|\vec{v}-\vec{p_i}\|} = \frac{1}{4\pi\epsilon_0} \sum_i \frac{q_i\,(\vec{v}-\vec{p_i})}{\|\vec{v}-\vec{p_i}\|^3}$$

Implementation of this model in Matlab is straightforward, and we can place the point charges $s_i$ in any way, to simulate various types and configurations of capacitors etc.. However, this method yields suboptimal results in term of precision of the calculated electric field. It is just an approximation of reality. But of course, the error can be made arbitrary small by increasing the number of point charges, as long as their distribution corresponds to reality.

## 2 Analytical model of electrical field

In this section, we will describe an analytical model that does not suffer from any approximation error. This model takes the ideas presented in previous section and assumes that there is infinite number of point charges, with charge that tends to zero for every single one of them. Then, we do the transition from sums to integrals. First, we will derive a general formula of $\vec{E}$ for arbitrary charge distribution.

Instead of a point charge $q_i$, we have now a continuous charge distribution $\rho : \mathbb{R}^3 \to \mathbb{R}$. A charge in volume $V \subseteq \mathbb{R}^3$ is then calculated by integration of the $\rho$ charge distribution: $Q_V = \iiint_V \rho(\vec{v}) dV$.

Now please note that for the sake of clarity and simplicity, we will abuse the integral notation a bit. We will use the integration operation for integrating vectors. Every time a vector output of the integration operation is expected, it means the integration is done for every vector component individually. This will greatly simplify the notation. So here is the general equation for a vector field generated by arbitrary charge distribution $\rho : \mathbb{R}^3 \to \mathbb{R}$.

$$\vec{E}(\vec{v}) = \iiint \frac{\rho(\vec{p})}{4\pi\epsilon_0 \left\|\vec{v}-\vec{p}\right\|^2} \cdot \frac{(\vec{v}-\vec{p})}{\left\|\vec{v}-\vec{p}\right\|} dP = \frac{1}{4\pi\epsilon_0} \iiint \frac{\rho(\vec{p})\,(\vec{v}-\vec{p})}{\left\|\vec{v}-\vec{p}\right\|^3} dP \tag{2}$$

Note this equation only holds for 3-dimensional charge distribution. If the charge is distributed on a 2-d plane in 3-dimensional space, then the integral has to be adapted accordingly, to integrate over a surface, not over a volume. The derivation is straightforward, I will not go in details here.

### 2.1 Application - calculating field of a capacitor plates (2-dimensional case)

At first, we will applicate the general formula (2) to a charged line in two dimensional plane. Assume that the line is a set $L = \{(x, c) | x \in [-d, d]\}$, for $c \in \mathbb{R}$ constant, and the appropriate charge density[1] $\rho$ on that line is constant.

Because the charge is located only on the line $L$, we will integrate only on this line.

$$E_x(v_x, v_y) = \frac{\rho}{2\pi\epsilon_0} \int_{-d}^{d} \frac{(v_x - x)}{(v_x - x)^2 + (v_y - c)^2} dx = \frac{\rho}{2\pi\epsilon_0} \left[ -\frac{1}{2} \log\left( (v_x - x)^2 + (c - v_y)^2 \right) \right]_{-d}^{d}$$

$$E_y(v_x, v_y) = \frac{\rho}{2\pi\epsilon_0} \int_{-d}^{d} \frac{(v_y - c)}{(v_x - x)^2 + (v_y - c)^2} dx = \frac{\rho}{2\pi\epsilon_0} \left[ \arctan\left( \frac{v_x - x}{c - v_y} \right) \right]_{-d}^{d}$$

[1] I use term *distribution* if the function varies across space, if it is constant I use term *charge density*

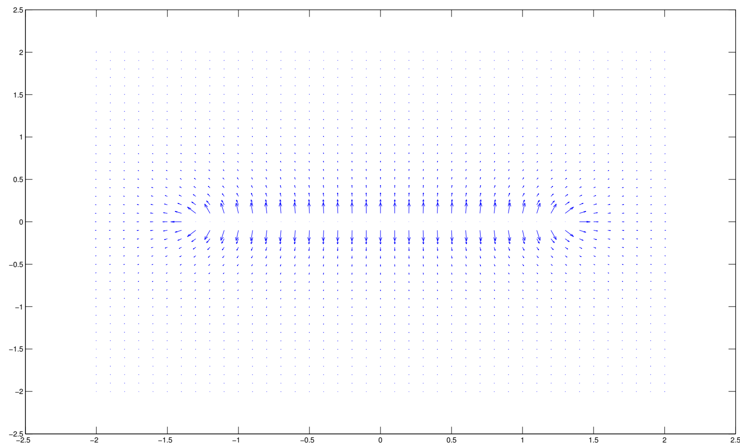


Figure 1: Single capacitor plate - Matlab plot of electrical field

Please note that electric field is a quantity that has physical meaning only in 3-dimensional space. If we want to consider a 2-dimensional case, slightly different formula for calculation of a point charge electric field has to be used:

$$E = \frac{q}{2\pi\epsilon_0 r}$$

This formula satisfies the same properties as the 3-dimensional formula would, if the z-dimension was invariant. To give better idea, imagine that the capacitor plate is extended to infinity in the z-dimension. Proof is straightforward (principle of superposition).

These formulas are then implemented in Matlab script to calculate the vector field precisely at any arbitrary point in space. A vector field plot is drawn. Please see the supplementary figures.

### 2.2 Application - calculating field of a capacitor plates (3-dimensional case)

The general formula for electric field caused by charge distributed on surface $S$ is:

$$\vec{E}(v_x, v_y, v_z) = \frac{1}{4\pi\epsilon_0} \iint_S \frac{\rho(s_x, s_y, s_z) \cdot (\vec{v} - \vec{s})}{\left(\sqrt{(v_x - s_x)^2 + (v_y - s_y)^2 + (v_z - s_z)^2}\right)^3} dS$$

We integrate over an arbitrary surface $S \subset \mathbb{R}^3$, which represents the charged surface. In the most general case, the charge density $\rho$ does not have to be constant and the surface can be of any shape. Presented equation holds for any

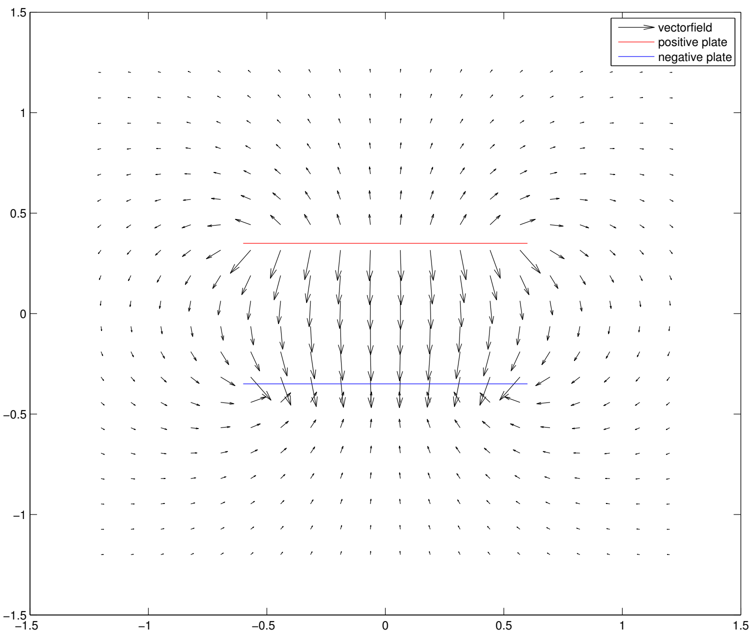


Figure 2: electrical field of a plate capacitor, 2D

arbitrary case. However, for now, we will consider a square shaped capacitor plates with constant charge density. The plate of such capacitor is a set $S = \{(x,y,c)|x \in [-d,d], y \in [-d,d]\}$ for $c \in \mathbb{R}$ constant, and the appropriate charge density $\rho$ for that surface is constant. Then, the equation for vector field is as follows:

$$E_x(v_x,v_y,v_z) = \frac{\rho}{4\pi\epsilon_0}\int_{-d}^{d}\int_{-d}^{d}\frac{(v_x-x)}{\left(\sqrt{(v_x-x)^2+(v_y-y)^2+(v_z-c)^2}\right)^3}dxdy$$

$$E_y(v_x,v_y,v_z) = \frac{\rho}{4\pi\epsilon_0}\int_{-d}^{d}\int_{-d}^{d}\frac{(v_y-y)}{\left(\sqrt{(v_x-x)^2+(v_y-y)^2+(v_z-c)^2}\right)^3}dxdy$$

$$E_z(v_x,v_y,v_z) = \frac{\rho}{4\pi\epsilon_0}\int_{-d}^{d}\int_{-d}^{d}\frac{(v_z-c)}{\left(\sqrt{(v_x-x)^2+(v_y-y)^2+(v_z-c)^2}\right)^3}dxdy$$

These integrals can be solved analytically. I used Matlab for symbolic integration of these formulas. The analytical form of the solution is quite complicated, so I am not presenting the solution here. However, it can be easily extracted from the Matlab program.

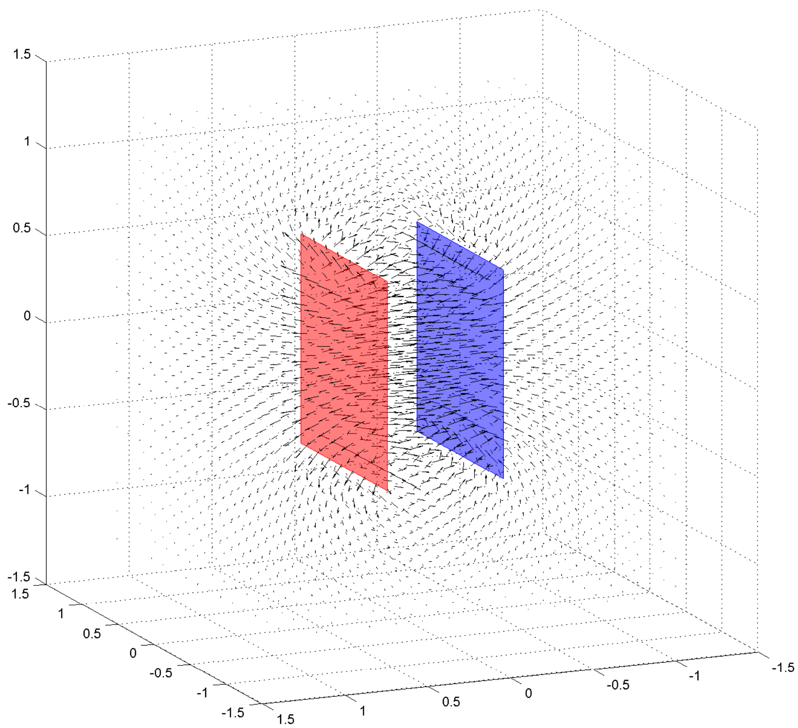


Figure 3: electrical field of a plate capacitor, 3D

# Part III
# Conclusion

In this paper, we presented a method how to calculate electrical field of any arbitrary charge distribution in space. First, we introduced discrete model of electrical field. Consequently, we have shown method for analytical calculation of electrical field. The analytical model were then implemented in Matlab; electrical field of square plate capacitor was calculated and plots were made in 2D and 3D.

Analytical method has considerable advantages over numerical solution of the problem. Most importantly, analytical expression for electric field yields absolutely accurate results. Furthermore, the calculation is fast even for huge models. Any numerical solution to this problem is suboptimal and its calculation takes much more processing power and time. However, for more complicated surfaces or charge distributions, the analytical formulas may not exist. In such case, numerical solution is the only way to go. I wrote this paper to show that for certain problems, the symbolic math toolbox in Matlab is much more suitable than the ordinary “numerical” way of calculation, which is intrinsic for Matlab.

# Appendinx - Matlab scripts

## Capacitor in two-dimensional space

Listing 1: Matlab script - field of a plate capacitor in 2D

```
%% Computation of electrical field of a plate capacitor,
    2D
% Ondøej Stanìk, 2012-12-14
% www.ostan.cz
%
% Maxwell Eqation class
% SRH Hochschule Heidelberg

close all  % close all figure windows that are open
clear all  % clear all the variables currently stored in
    memory
clc        % clear the commands in the command window

syms x y vx vy

e0 = 8.854187817620E-12; % permitivity of free space

plateLength = 1.2; % length of the plate in meters
chargeDensity = 1; % line charge density of capacitor
    plates (constant, 2-dimensional case), in [Q/m]
gridResolution = 20; % resolution of grid
plateDistance = 1; % distance of capacitor plates in
    meters

% normalization constant that gives our calculations
    physical meaning in
% terms of SI units..
physicalConstant = 1/(2*pi*e0); % circumference of a
    circle

% the analytic expression for vector x-coordinate element
      at coordinates
```

```
% (vx,vy) - indefinite integral
Ex = int((vx-x)/((vx-x)^2+(vy-plateDistance/2)^2),x) *
    physicalConstant * chargeDensity;

% the analytic expression for y-vector y-coordinate
    element at coordinates (vx,vy)
Ey = int((vy-plateDistance/2)/((vx-x)^2+(vy-plateDistance
    /2)^2)) * physicalConstant * chargeDensity;

ExDefinite = subs(Ex,x,plateLength/2) - subs(Ex,x,-
    plateLength/2)
EyDefinite = subs(Ey,x,plateLength/2) - subs(Ey,x,-
    plateLength/2)

% substitution of a specific numbers to symbolic
    expressions Ex, Ey.
ExS = @(vxs,vys) subs(ExDefinite,[vx,vy],[vxs,vys]);
EyS = @(vxs,vys) subs(EyDefinite,[vx,vy],[vxs,vys]);

% generate vector coordinates
X = linspace(-plateLength,plateLength,gridResolution);
Y = linspace(-plateLength,plateLength,gridResolution);

% initialize 2-dimensional array to store vector field
VX = zeros(gridResolution,gridResolution);
VY = zeros(gridResolution,gridResolution);

% calculate the vector field for one plate
for x = 1:1:gridResolution
    for y = 1:1:gridResolution
        VX(y,x) = ExS(X(x),Y(y));
        VY(y,x) = EyS(X(x),Y(y));

        percentFinished = 100*((x-1)/gridResolution + (y
            -1)/gridResolution^2)
    end
end

% compute the field for negative capacitor plate
% simple vectorfield transformation: just flip the
    vectorfield around y-axis and reverse sign
VX2 = - flipud(VX);
VY2 = flipud(VY);

% compute the field supperposition of the negative and
    positive capacitor plate
```

```
VXS = VX + VX2;
VYS = VY + VY2;

% plot vector field in 2D:
figure;
quiver(Y,X,VXS,VYS, 'k');

% plot the capacitor plates
hold on;
plot([-plateLength/2, plateLength/2],[plateDistance/2,
    plateDistance/2], 'r-');
plot([-plateLength/2, plateLength/2],[-plateDistance/2, -
    plateDistance/2], 'b-');
legend('vectorfield','positive␣plate','negative␣plate');
```

## Capacitor in three-dimensional space

Listing 2: Matlab script - field of a plate capacitor in 3D

```
%% Computation of electrical field of a plate capacitor,
    3D
% Ondøej Stanìk, 2012-12-29
% www.ostan.cz
%
% Maxwell Eqation class
% SRH Hochschule Heidelberg

close all  % close all figure windows that are open
clear all  % clear all the variables currently stored in
    memory
clc        % clear the commands in the command window

syms x y z vx vy vz

e0 = 8.854187817620E-12; % permitivity of free space

plateLength = 1.2; % length of the plate in meters
chargeDensity = 1; % surface charge density of capacitor
    plates (constant), in [Q/m^2]
gridResolution = 8; % resolution of grid
plateDistance = 0.7; % distance of capacitor plates in
    meters

% normalization constant that gives our calculations
    physical meaning in
% terms of SI units..
```

```
physicalConstant = 1/(4*pi*e0);

% the analytic expression for vector x-coordinate element
    at coordinates (vx,vy,vz)
Ex = int((vx-x)/((((vx-x)^2+(vy-y)^2+(vz-plateDistance/2)
   ^2)^(3/2)),x) * physicalConstant * chargeDensity;
Ex2 = subs(Ex,x,plateLength/2) - subs(Ex,x,-plateLength
   /2);
Ex3 = int(Ex2,y);
ExDefinite = subs(Ex3,y,plateLength/2) - subs(Ex3,y,-
   plateLength/2)

% the analytic expression for vector y-coordinate element
    at coordinates (vx,vy,xz)
Ey = int((vy-y)/((((vx-x)^2+(vy-y)^2+(vz-plateDistance/2)
   ^2)^(3/2)),x) * physicalConstant * chargeDensity;
Ey2 = subs(Ey,x,plateLength/2) - subs(Ey,x,-plateLength
   /2);
Ey3 = int(Ey2,y);
EyDefinite = subs(Ey3,y,plateLength/2) - subs(Ey3,y,-
   plateLength/2)

% the analytic expression for vector z-coordinate element
    at coordinates (vx,vy,xz)
Ez = int((vz-plateDistance/2)/((vx-x)^2+(vy-y)^2+(vz-
   plateDistance/2)^2)^(3/2),x) * physicalConstant *
   chargeDensity;
Ez2 = subs(Ez,x,plateLength/2) - subs(Ez,x,-plateLength
   /2);
Ez3 = int(Ez2,y);
EzDefinite = subs(Ez3,y,plateLength/2) - subs(Ez3,y,-
   plateLength/2)

% substitution of a specific numbers to symbolic
   expressions Ex, Ey.
ExS = @(vxs,vys,vzs) subs(ExDefinite,[vx,vy,vz],[vxs,vys,
   vzs]);
EyS = @(vxs,vys,vzs) subs(EyDefinite,[vx,vy,vz],[vxs,vys,
   vzs]);
EzS = @(vxs,vys,vzs) subs(EzDefinite,[vx,vy,vz],[vxs,vys,
   vzs]);

% generate vector coordinates
X = linspace(-plateLength,plateLength,gridResolution);
Y = linspace(-plateLength,plateLength,gridResolution);
Z = linspace(-plateLength,plateLength,gridResolution);
```

```
% initialize 3-dimensional array to store vector field
VX = zeros(gridResolution,gridResolution,gridResolution);
VY = zeros(gridResolution,gridResolution,gridResolution);
VZ = zeros(gridResolution,gridResolution,gridResolution);

% calculate the vector field for one plate
for x = 1:1:gridResolution
    for y = 1:1:gridResolution
        for z= 1:1:gridResolution
          VX(z,y,x) = ExS(X(x),Y(y),Z(z));
          VY(z,y,x) = EyS(X(x),Y(y),Z(z));
          VZ(z,y,x) = EzS(X(x),Y(y),Z(z));

          percentFinished = 100*((x-1)/gridResolution + (
              y-1)/gridResolution^2 + (z-1)/gridResolution
              ^3)
        end
    end
end

% calculate the field of negative plate, take advantage
    of symetricity
VX2 = flipdim(VX,1) * -1;
VY2 = flipdim(VY,1) * -1;
VZ2 = flipdim(VZ,1) * 1;

% superpose the field of negative and positive plate
VXS = VX + VX2;
VYS = VY + VY2;
VZS = VZ + VZ2;

% plot vector field in 3D:
[XX, YY, ZZ] = meshgrid(X,Y,Z);

figure;
quiver3(XX,YY,ZZ,VYS,VZS,VXS,'k');

% plot the capacitor plates
hold on
X = [-plateLength/2 plateLength/2; -plateLength/2
    plateLength/2;];
Y = [plateLength/2 plateLength/2; -plateLength/2 -
    plateLength/2;];
Z = ones(size(X))*plateDistance/2;
mesh(Y,Z,X,'EdgeColor','red','FaceAlpha',0.5,'FaceColor',
```

```
    'red');
mesh(Y,Z*-1,X,'EdgeColor','blue','FaceAlpha',0.5,'
    FaceColor','blue');
legend('vectorfield','positive plate','negative plate');
```